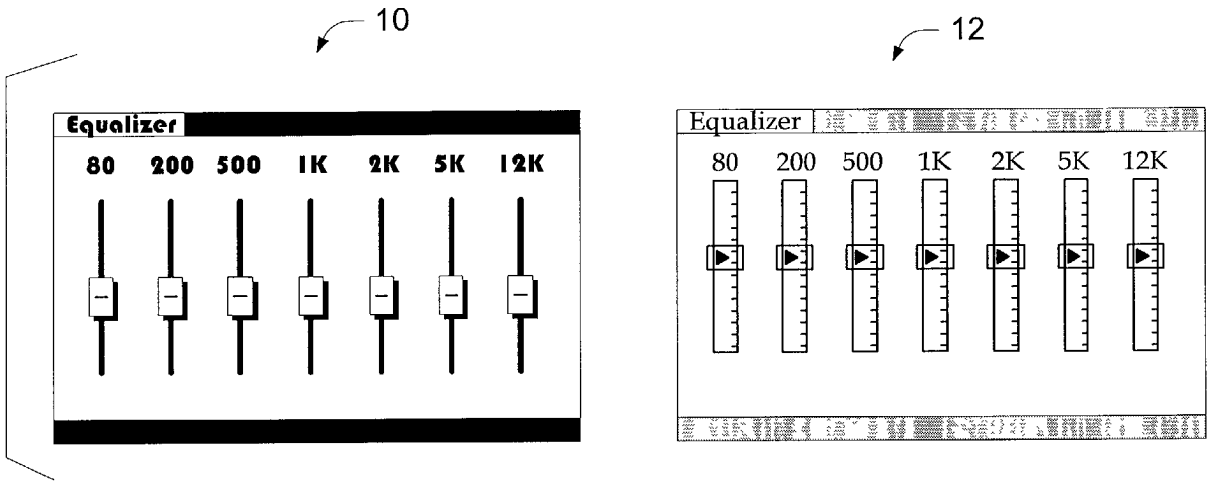
10
12
Equalizer
80 200 500 1K 2K 5K 12K
Equalizer
80 200 500 1K 2K 5K 12K

Fig. 1

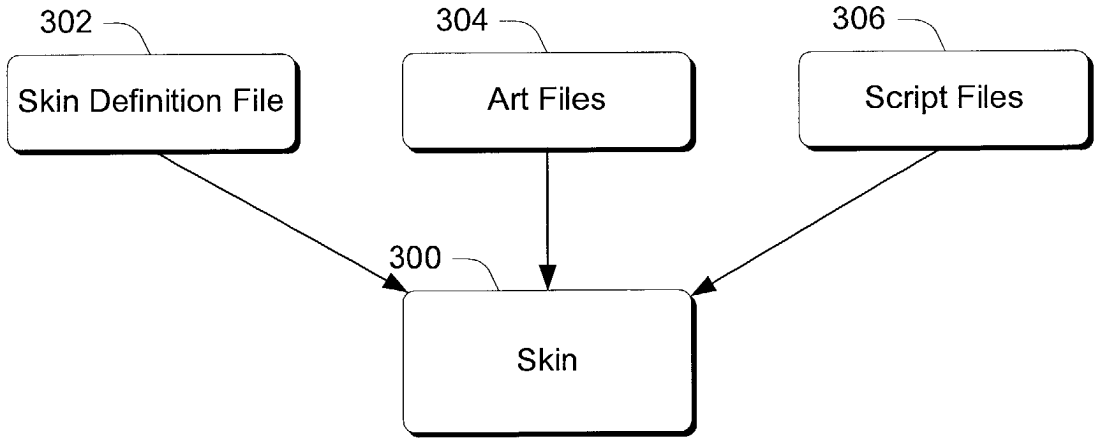
302
304
306
Skin Definition File
Art Files
Script Files
300
Skin

Fig. 3

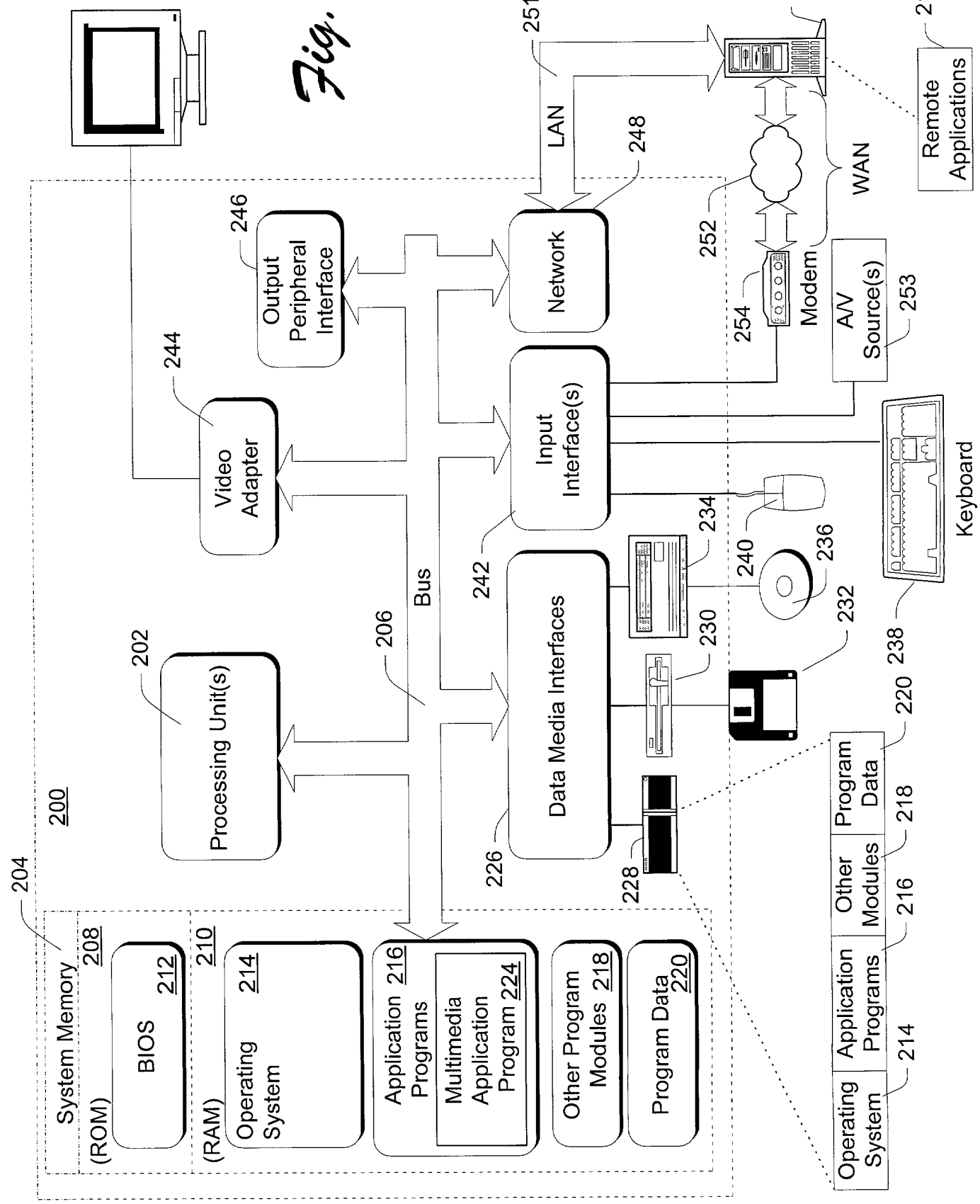
200
204
System Memory
(ROM)
208
BIOS
212
(RAM)
210
Operating System
214
Application Programs
216
Multimedia Application Program
224
Other Program Modules
218
Program Data
220
Processing Unit(s)
202
206
Bus
Video Adapter
244
Output Peripheral Interface
246
Network
248
LAN
251
WAN
252
Modem
254
A/V Source(s)
253
Input Interface(s)
242
Data Media Interfaces
226
228
230
232
234
236
238
240
Keyboard
250
Remote Applications
216
Operating System
214
Application Programs
216
Other Modules
218
Program Data
220

Fig. 2

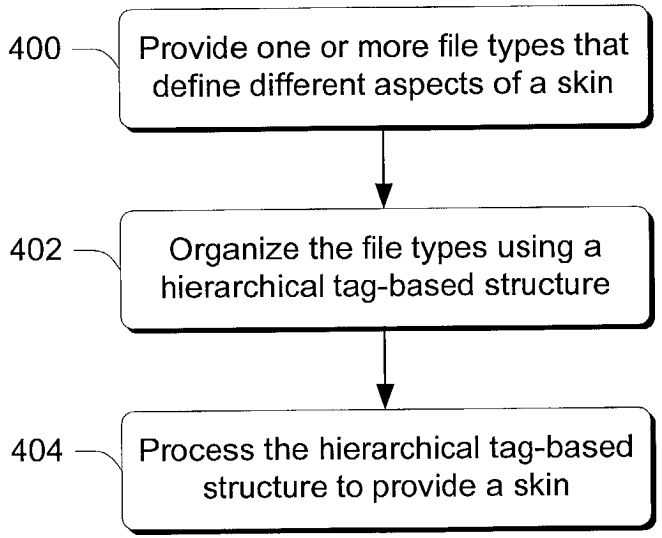
400
Provide one or more file types that define different aspects of a skin
402
Organize the file types using a hierarchical tag-based structure
404
Process the hierarchical tag-based structure to provide a skin

Fig. 4

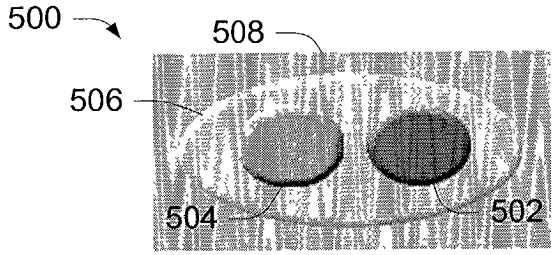

Fig. 5

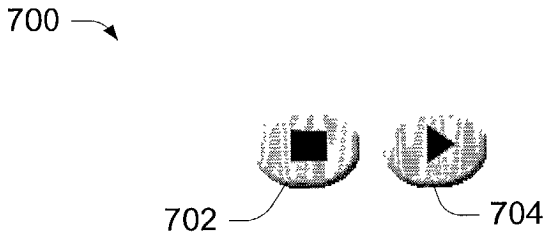

Fig. 7

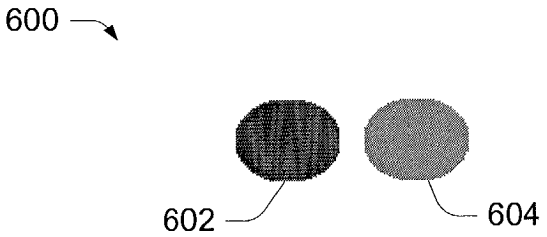

Fig. 6

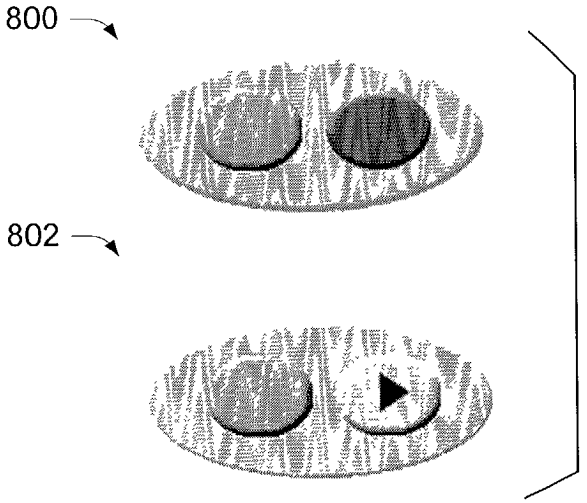

Fig. 8

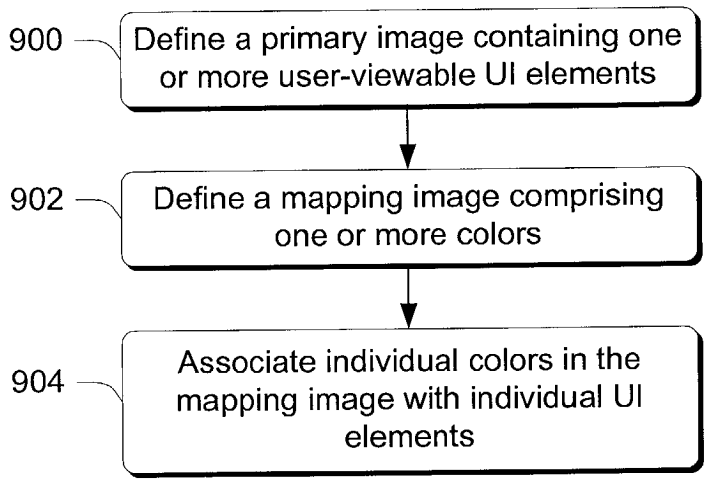

Fig. 9

1000

```
<THEME>
  <VIEW
    clippingColor = "#CCCC00"
    backgroundImage = "background.bmp"
    titleBar = "false">

    <BUTTONGROUP
      mappingImage = "map.bmp"
      hoverImage = "hover.bmp">

    <PLAYELEMENT
      mappingColor = "#00FF00"/>

    <STOPELEMENT
      mappingColor = "#FF0000"/>

    </BUTTONGROUP>
  </VIEW>
</THEME>
```

Fig. 10

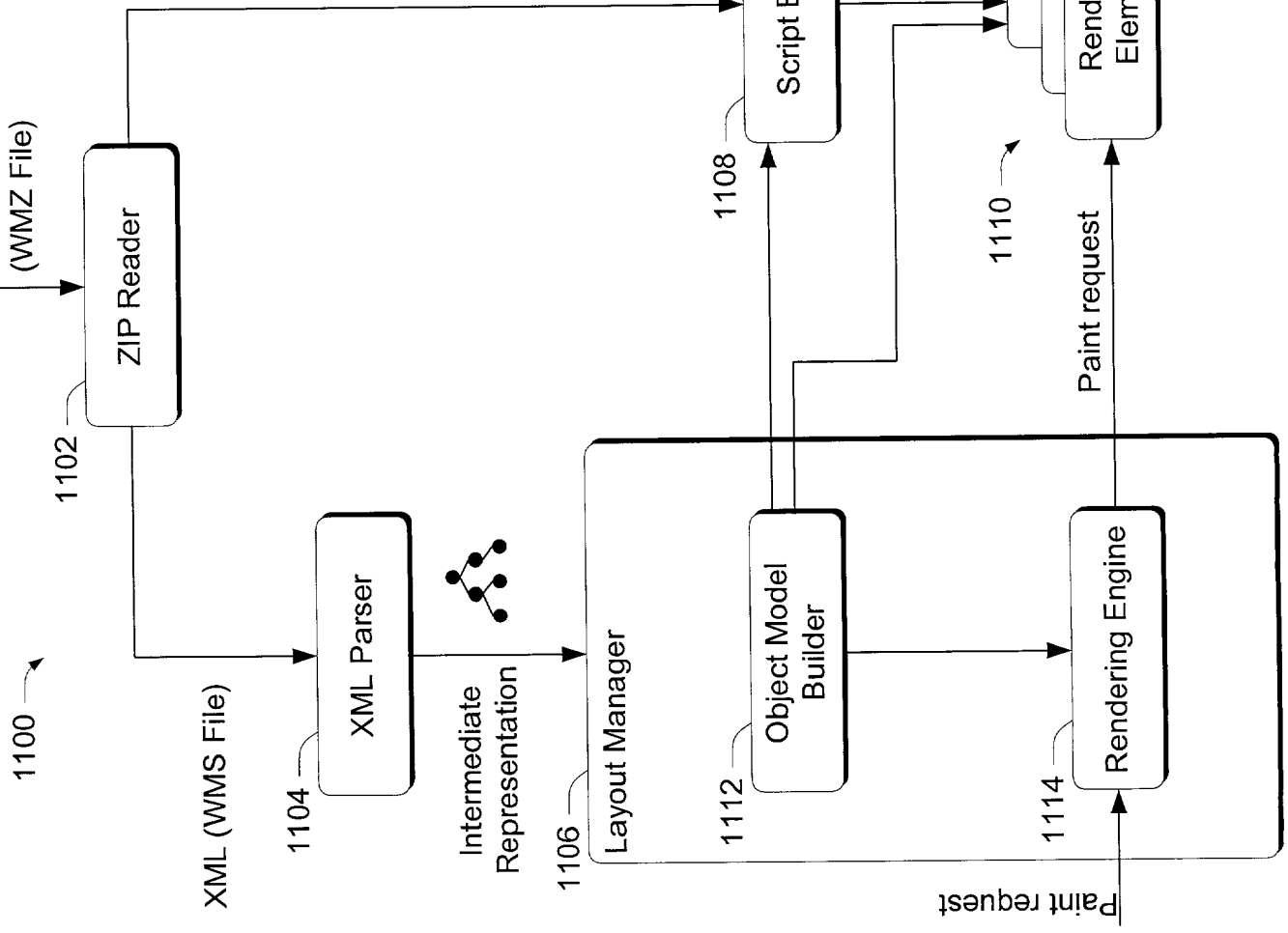
1100
Skin Archive
(WMZ File)
ZIP Reader
1102
XML (WMS File)
XML Parser
1104
Intermediate
Representation
1106
Layout Manager
Object Model
Builder
1112
Rendering Engine
1114
Paint request
Script Engine
1108
1110
Rendering
Elements

Fig. 11

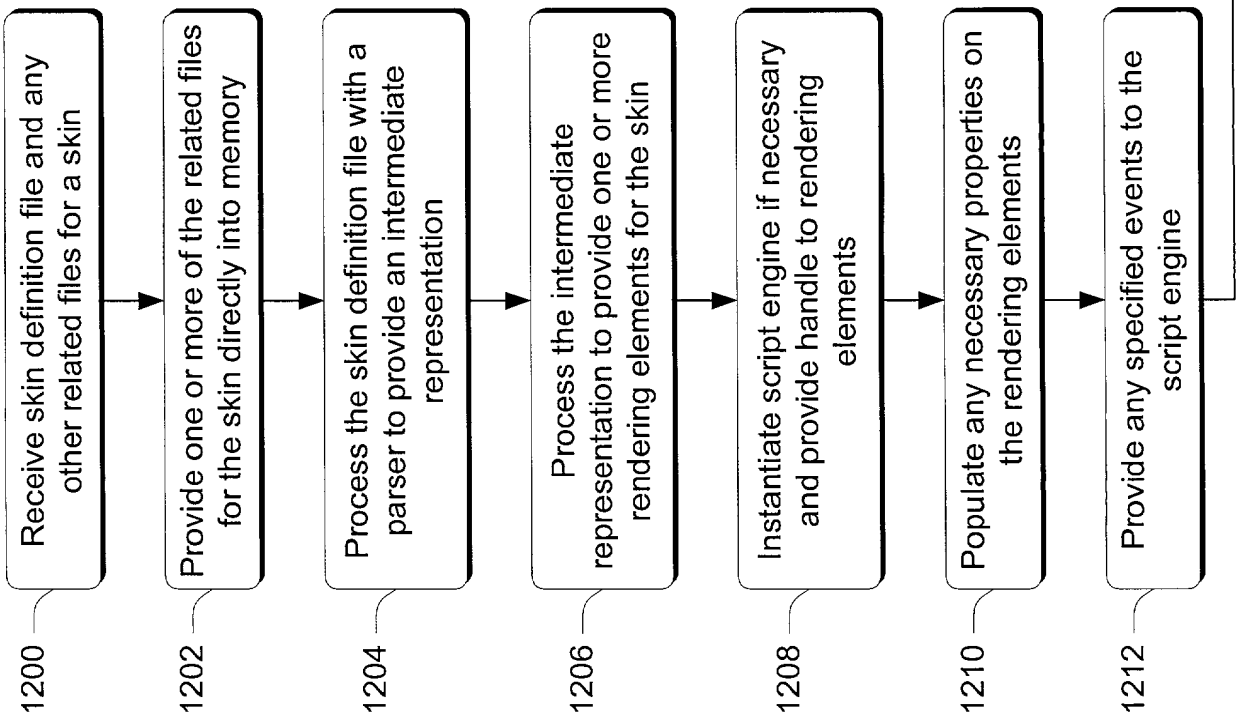
1200
Receive skin definition file and any other related files for a skin
1202
Provide one or more of the related files for the skin directly into memory
1204
Process the skin definition file with a parser to provide an intermediate representation
1206
Process the intermediate representation to provide one or more rendering elements for the skin
1208
Instantiate script engine if necessary and provide handle to rendering elements
1210
Populate any necessary properties on the rendering elements
1212
Provide any specified events to the script engine
1214
Provide any necessary files to their associated rendering elements
1216
Provide any necessary script files to the script engine

Fig. 12

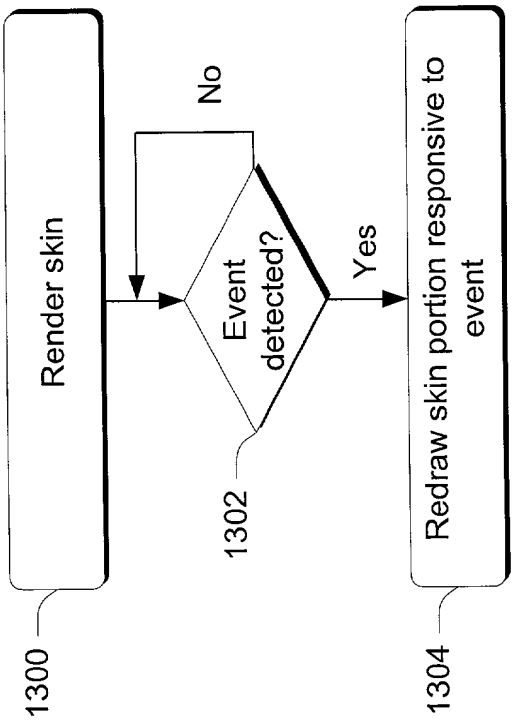
Render skin
Event detected?
No
Yes
Redraw skin portion responsive to event
1300
1302
1304

Fig. 13

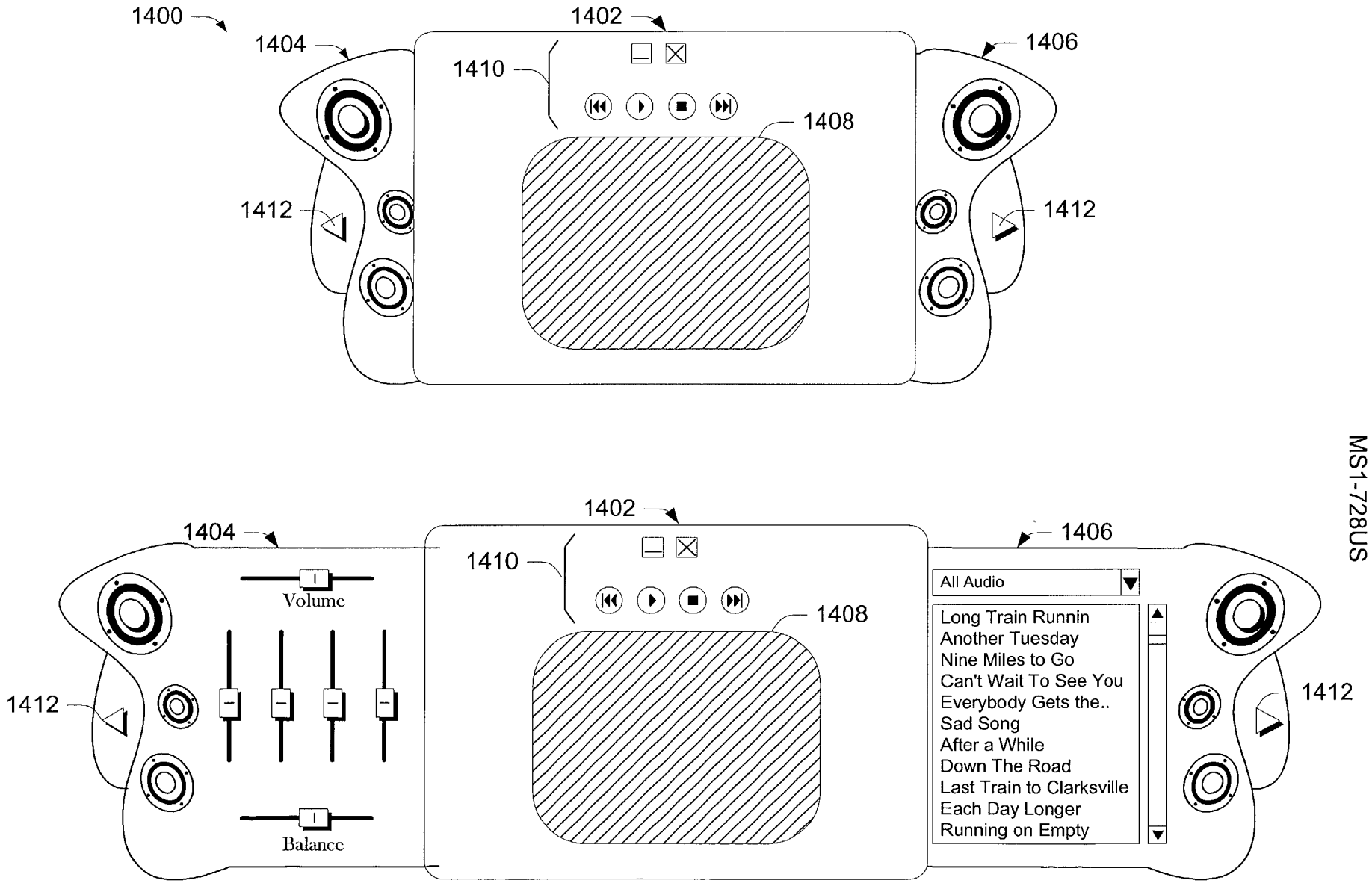
1400
1402
1404
1406
1410
1408
1412
1412
1402
1404
1406
1410
1408
Volume
Balance
1412
All Audio
Long Train Runnin
Another Tuesday
Nine Miles to Go
Can't Wait To See You
Everybody Gets the..
Sad Song
After a While
Down The Road
Last Train to Clarksville
Each Day Longer
Running on Empty

Fig. 14

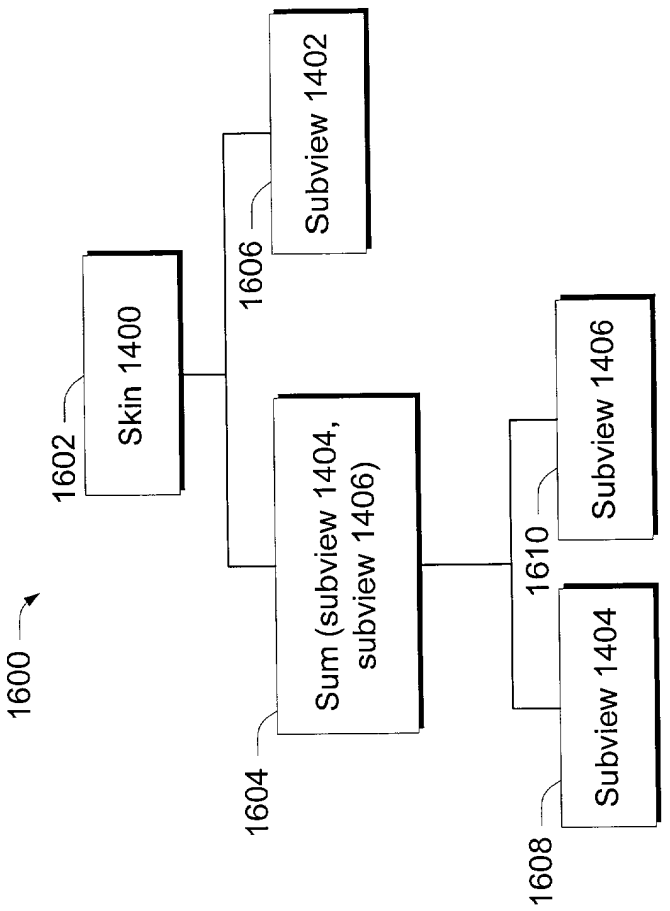
1600
1602
Skin 1400
1604
Sum (subview 1404, subview 1406)
1606
Subview 1402
1608
Subview 1404
1610
Subview 1406

Fig. 16

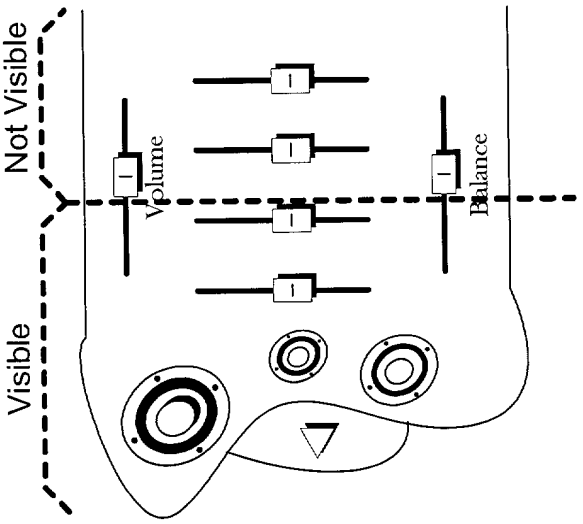
1404
Visible
Not Visible
Volume
Balance

Fig. 15

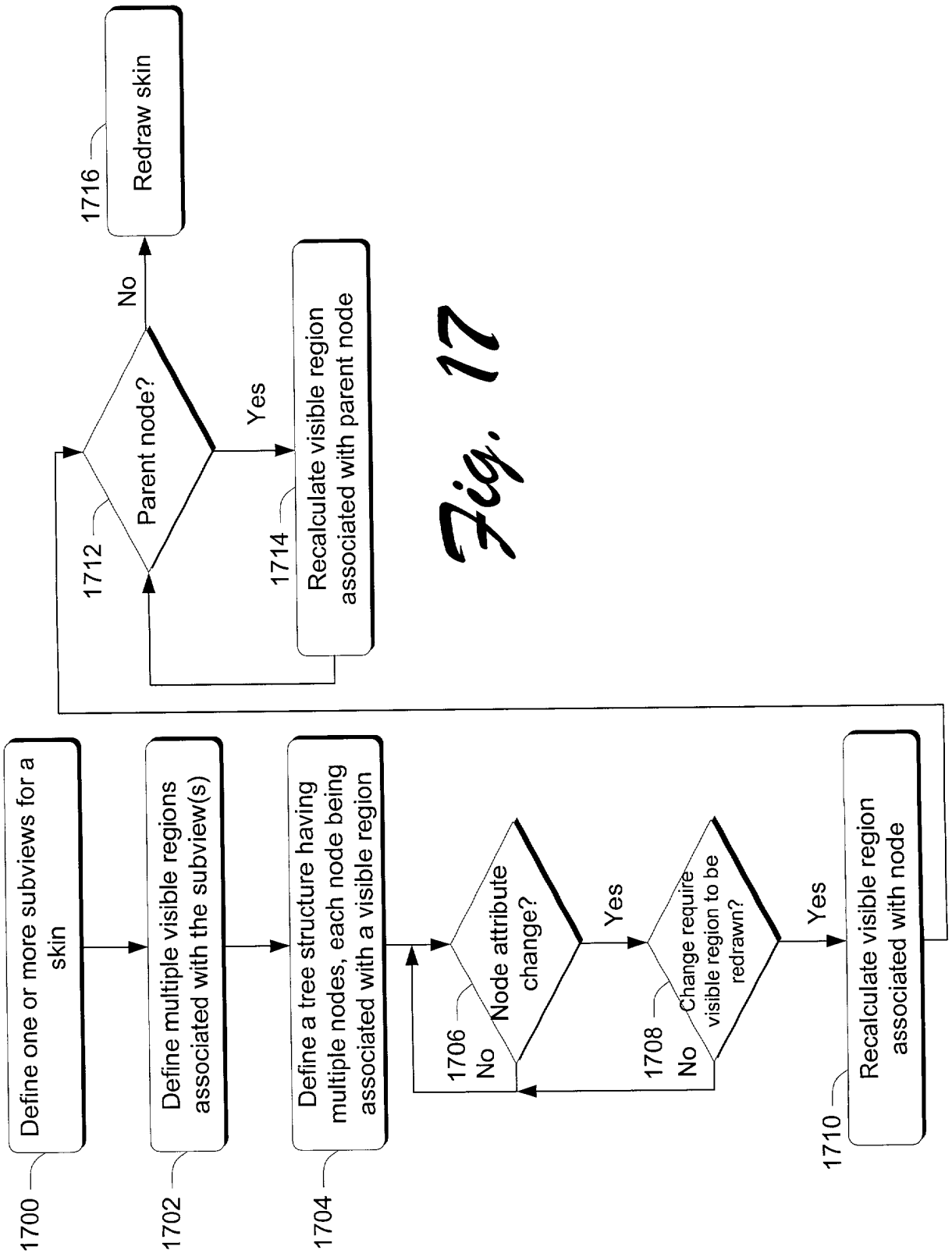
1700
Define one or more subviews for a skin
1702
Define multiple visible regions associated with the subview(s)
1704
Define a tree structure having multiple nodes, each node being associated with a visible region
1706
Node attribute change?
No
Yes
1708
Change require visible region to be redrawn?
No
Yes
1710
Recalculate visible region associated with node
1712
Parent node?
No
Yes
1714
Recalculate visible region associated with parent node
1716
Redraw skin

Fig. 17

```
<THEME>
  <VIEW>
      <PLAYER>
          <SETTINGS>
              volume_onchange="volumeslider.value = volume;" >
          </SETTINGS>
      </PLAYER>
      <SLIDER>
          id="volumeslider"
          min="0"
          max="100"
          onpositionchange="player.settings.volume = value;" >
      </SLIDER>
  </VIEW>
</THEME>
```

Fig. 18

```
<THEME>
  <VIEW>
      <SLIDER>
          min="0"
          max="100"                    1900
          value="wmpprop:player.settings.volume"
          onpositionchange="player.settings.volume = value;" >
      </SLIDER>                        1902
  </VIEW>
</THEME>
```

Fig. 19

```
<THEME>
  <VIEW>
      <BUTTON
          id=play
          visible="TRUE" />
      <BUTTON
          id=pause
          visible="wmpprop:play.visible" />
  </VIEW>
</THEME>
```

Fig. 20

PUBLISHED SPECIFICATION

skin.wms

```
<THEME>
  <VIEW>
        <PLAYER>
              playstatechange="EnablePlayButton();"
              openstatechange="EnablePlayButton();"
        </PLAYER>
        <PLAYBUTTON
              id="play"
              image="play.bmp"
              disabledimage="play_disabled.bmp"
              onpositionchange="player.settings.volume = value;"  >
        </PLAYBUTTON>
  </VIEW>
</THEME>
```

skin.js

```
function EnablePlayButton ()
{
        play.enabled = (player.playState != wmppsPlaying)  &&
                       (player.openState == wmposMediaOpen);
}
```

Fig. 21

Docket No. 328262.01

```
<THEME>
  <VIEW>
      <PLAYBUTTON
          image="play.bmp"
          disabledimage="play_disabled.bmp"
          enabled="wmpenabled:player.controls.play();"  >
      </PLAYBUTTON>
  </VIEW>
</THEME>
```

Fig. 22

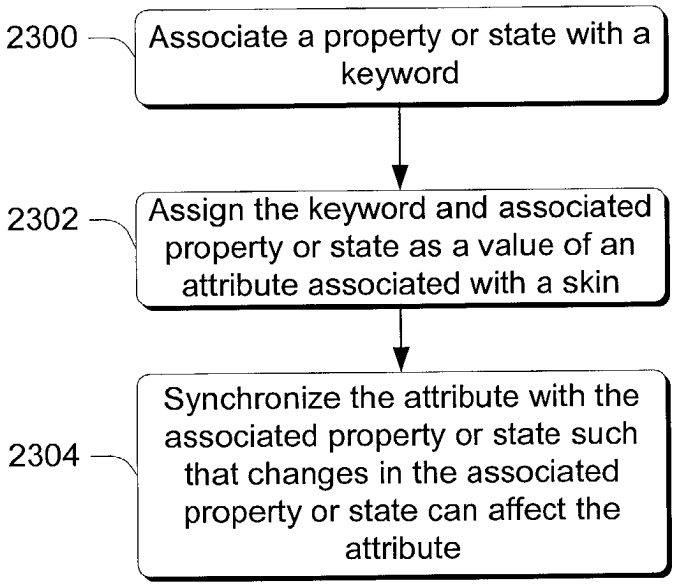

Fig. 23

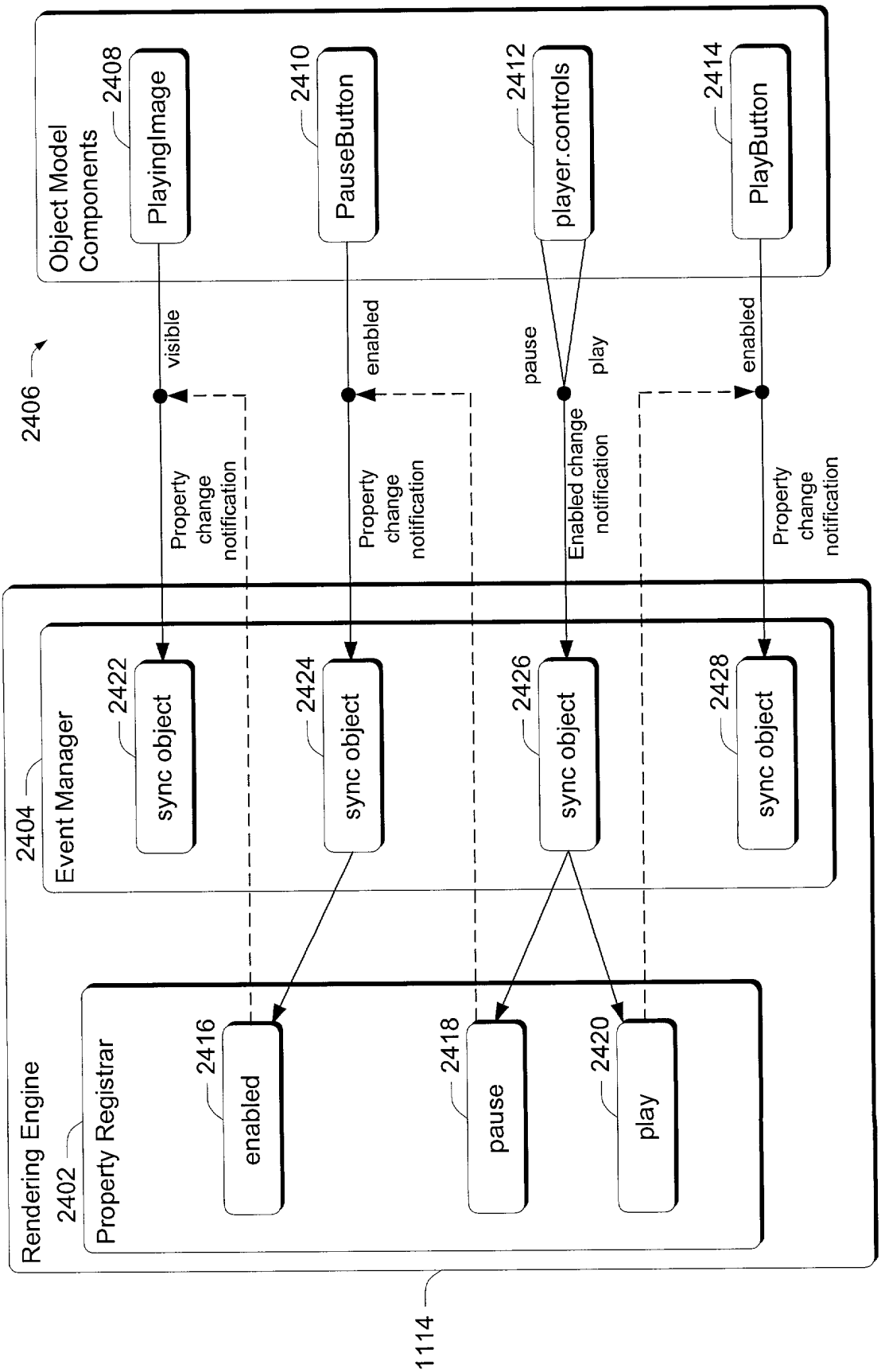
Object Model Components
2408
PlayingImage
2410
PauseButton
2412
player.controls
2414
PlayButton
2406
visible
enabled
pause
play
enabled
Property change notification
Property change notification
Enabled change notification
Property change notification
Event Manager
2404
2422
sync object
2424
sync object
2426
sync object
2428
sync object
Rendering Engine
2402
Property Registrar
2416
enabled
2418
pause
2420
play
1114

Fig. 24

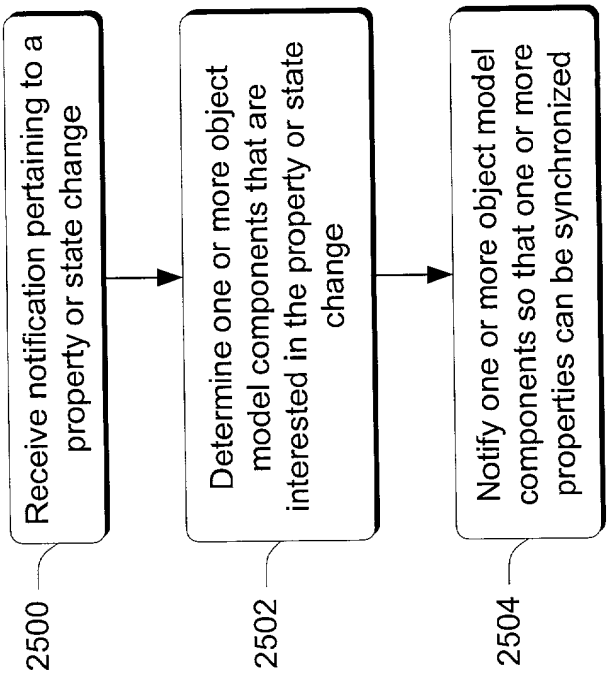
Receive notification pertaining to a property or state change
Determine one or more object model components that are interested in the property or state change
Notify one or more object model components so that one or more properties can be synchronized
2500
2502
2504

Fig. 25